(1)

A HEURISTIC ALGORITHM FOR THE FACILITIES LAYOUT PROBLEM

by

Bharat K. Kaku[*]
Thomas E. Morton[**]
and
Gerald L. Thompson[**]

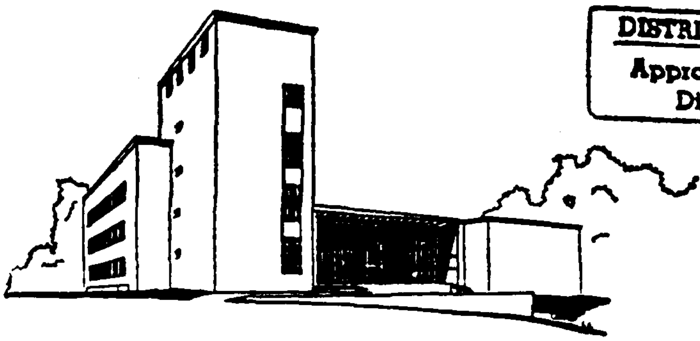
May 9, 1988

Carnegie Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



①

Management Science Research Report MSRR 544

A HEURISTIC ALGORITHM FOR THE FACILITIES LAYOUT PROBLEM

by

Bharat K. Kaku*
Thomas E. Morton**
and
Gerald L. Thompson**

May 9, 1988

* Department of Management Information Systems
The University of Arizona
Tucson, Arizona 85721

** Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared in part as part of the activities of the Management Science Research Group, Carnegie Mellon University, under Contract No. N00014-85-K-0198 NR 047-048 with the Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

Management Sciences Research Group
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

This paper presents a heuristic algorithm for solving the facilities layout problem. The basic approach is the combination of a constructive method with exchange procedures, used repetitively. The constructive heuristic uses alternate costs, obtained in the process of computing the Gilmore-Lawler bounds, as the criterion for choosing the next assignment. Different partial solutions, to be used as starting points for multiple application of the constructive procedure, are obtained by development of a restricted breadth-first branch and bound tree. Computational results show that the method compares favourably with two competing procedures from the literature in finding solutions within 0.40% of the best known solutions for well known problems. Computing times are reasonable for problems with as many as 36 facilities. We also present a new best known solution for one version of the Steinberg problem, found in the process of experimentation.

1 Introduction

The quadratic assignment problem formulation applies to a wide and diverse range of problems: the location of interdependent plants or facilities, the layout of interacting departments in an office building, the location of medical facilities in a hospital, the location of indicators and controls on a control panel or in a control room, the backboard wiring problem in the design of computers and other electronic equipment, the traveling salesman problem, and the production sequencing problem with dependent setup times. These problems are similar in structure to the classical linear assignment problem of assigning indivisible facilities to discrete locations, but are more complicated because the objective function contains terms that are quadratic in the decision variables, arising due to the interdependence of facilities. The quadratic assignment problem (QAP) is known to be an NP-complete problem and no known solution method is capable of solving problems with 15 or more facilities optimally in reasonable time. Consequently, there is a need for heuristic methods that provide good sub-optimal solutions. In the literature, the QAP is generally discussed in terms of the facilities layout problem (FLP), which is perhaps the best known application. Given the fact that solution methods can be applied to different problems by simply redefining coefficients and variables, this practice is adopted, without loss of generality.

The heuristic solution procedure for the facilities layout problem presented in this paper is a combination of a constructive technique for obtaining complete solutions, and exchange-improvement routines for improving them. The constructive procedure is based on the use of alternate costs, choosing as the next assignment to be added to the present partial solution the assignment with the largest alternate cost. The procedure can then be repeated on the augmented partial solution, until all facilities have been assigned. To improve the quality of the solution found, it was deemed necessary to construct several solutions, attempt to improve them through exchange procedures, and retain the best solution arrived at. Towards this end, a method was devised to provide several partial assignments as starting points for the construction procedure, which then completes them, arriving at different complete solutions. The objective was achieved through development of the first few levels of a restricted breadth-first type decision tree. The nodes of this tree contain distinct partial assignments that provide the starting points for repeated application of the constructive technique. The final phase of the algorithm consists of exchange improvement of these solutions.

1.1 Mathematical Formulation

We define the following matrices and notation:

$\mathcal{A}$ = $\|a_{ij}\|$ is the fixed cost matrix, where a_{ij} is the fixed linear cost of installing facility i at location j

$\mathcal{F}$ = $\|f_{ij}\|$ is the intensity or flow matrix, where f_{ij} is the cost per unit distance of transporting the flow from facility i to facility j

$\mathcal{D}$ = $\|d_{ij}\|$ is the distance matrix, where d_{ij} is the distance from location i to location j

$\rho(i)$ is the location to which facility i is assigned

With these definitions, the facilities layout problem (FLP) is to find a one-to-one mapping of the set of facilities ($\mathcal{N} = \{1, 2, \ldots, n\}$) into the set of locations ($\mathcal{N}$) so as to:

$$\text{Minimize} \sum_{i \in \mathcal{N}} a_{i\rho(i)} + \sum_{i,j \in \mathcal{N}} f_{ij} d_{\rho(i)\rho(j)} \tag{1}$$

2 Review of Existing Algorithms

Optimal algorithms for the facilities layout problem can be classified under two categories:

- Implicit Enumeration, including single-assignment algorithms proposed by Gilmore [10] and Lawler [17]; and pair-assignment algorithms proposed by Land [16] and Gavett and Plyter [9].
- Linearizations, such as those proposed by Lawler [17], Bazaraa and Sherali [3], Kaufman and Broeckx [15], Balas and Mazzola [1], and Kaku and Thompson [14].

Various heuristic procedures have been devised for the facilities layout problem and can be grouped under four major categories:

- limited enumeration
- constructive procedures
- heuristic solutions to linearized problems
- improvement procedures

For more detailed reviews and experimental comparisons of heuristic techniques, see Nugent, Vollmann and Ruml [20], Hanan and Kurtzberg [12], Ritzman [22], Liggett [18], Burkard and Stratmann [6], Picone and Wilhelm [21], and Kaku and Thompson [13].

2.1 Hybrid Algorithms for the FLP

Hanan and Kurtzberg [12] suggest in their paper that superior performance (in terms of solution quality and computing time) could be achieved by combining a constructive method with an iterative-improvement procedure. A constructive heuristic is one that, starting from any partial assignment (perhaps empty), chooses the next assignment to be made by evaluating the assignment of unassigned facilities to free locations. This process is repeated until a complete solution has been constructed. Such a coupling of improvement routines with other techniques—constructive and otherwise—has proved to be beneficial. In this section we review the literature on hybrid algorithms for the FLP. The term hybrid here is used to refer to methods that employ some technique to construct or obtain complete assignments for the FLP and then attempt to improve these by applying exchange routines. The best results mentioned in the literature have been achieved by algorithms using such a two-phase approach, and we provide a brief discussion of the better ones, along with sample results in terms of solution quality and computation time.

The computation times for all the algorithms discussed in this section are quoted from the respective papers without any attempt to normalize for differences in computers and programming languages used. However, the computer and language are mentioned wherever this information is available. The problems for this chapter are taken from the following sources. The NVR problems are randomly generated problems from a paper by Nugent, Vollmann and Ruml [20], where the number at the end refers to the size of the problem. The name STEIN is used to refer to the backboard wiring problem presented by Steinberg [23], where 34 components are required to be placed in a 9×4 rectangular grid; necessitating the inclusion of two dummy components. Three options are available in the way distances between locations are measured. RD stands for rectilinear distances, ED for euclidean distances and SED represents squared euclidean distances. The ELSHAFEI problem [7] is a 19 department hospital layout problem.

Burkard and Stratmann [6] presented two combination methods. The first applies the Gaschutz and Ahrens [8] graph-theoretic approach to several (50-80) random assignments to obtain solutions which can be improved by exchange procedures (ALG1). The exchanges are carried out by a sophisticated improvement routine called 'VERBES' which is described below. The second method employs a restricted tree search coupled with VERBES (ALG2) as follows:

1. Using a branch and bound procedure based on the Gilmore-Lawler method with a time limit, determine a suboptimal solution.
2. Improve this solution by VERBES.
3. If an improvement is made in (2), determine the smallest level k in the search tree that was affected by the exchange algorithm.
4. Restart the branch and bound process at this level k, ensuring that assignments already evaluated are not considered again. Return to step (2).

The package VERBES operates as follows:

1. Apply a pair exchange algorithm $n/3$ times (where n is the number of facilities), using arbitrary order of exchanges.
2. Store the three best solutions found in (1).
3. Apply a triple exchange algorithm to all three solutions from (2). These exchanges are carried out in a fixed order based on the actual assignments.
4. If an improvement is made in (3), repeat the pair exchange and triple exchange, both in a fixed order determined as in (3), until one of them does not produce a reduction in costs. (For details on the method for determining the fixed order referred to above, see pages 141-142 of [6].)

A hybrid procedure due to Liggett [19] makes use of a constructive technique based on the Graves-Whinston [11] algorithm. The Graves-Whinston algorithm provides a means of computing an expected value for the completion of any partial assignment, using statistical properties. In the constructive phase of Liggett's method, the facility-location pair offering the smallest expected value for a complete assignment is chosen as the next assignment to be fixed, just as the pair with the smallest lower bound would be chosen in a depth first search tree. To obtain multiple alternate solutions for application of an improvement procedure, the method incorporates a limited backtracking mechanism as follows. At *early* levels, if the second best expected cost for the assignment of the *chosen* facility is within 0.50% of the smallest expected cost, this node is saved for exploration of an alternate route through the partial decision tree. Such assignments are termed "critical junctions" and by backtracking to them, alternate solutions can be generated. Thus the method, while termed a constructive procedure, has certain features of limited tree search. The complete solutions obtained are subjected to a simple pair exchange improvement procedure. See Table 1 for results.

Table 1: Results for the Burkard-Stratmann and Liggett algorithms

PROBLEM	Burkard-Stratmann[a]				Liggett[b]	
	ALG1		ALG2			
NVR20	1297[c]	3:00[d]	1287	1:26	1308	0:01
NVR30	3089	10:00	3079	6:51	3103	0:01
STEIN RD	4804	20:00	4807	13:59	-	-
STEIN ED	4132.97	20:00	4132.29	18:09	4141	-
STEIN SED	7926	20:00	8109	20:54	-	-

[a]Fortran programs on a CDC Cyber 72/76.

[b]Fortran programs on an IBM 360/91.

[c]Objective function value

[d]Computation time (mins:secs)

The two methods of Burkard and Stratmann (ALG1 and ALG2) are comparable in terms of solution quality and computation times but the second is universally applicable as opposed to the Gaschutz and Ahrens method which assumes a symmetric distance matrix and requires the locations to lie in a rectangle. The computation times for the Gaschutz-Ahrens plus VERBES method are approximate times mentioned in [6]. The Liggett algorithm has a distinct advantage over the others discussed in this section in terms of computation times, which are an order of magnitude smaller. The quality of the solutions found is, however, comparatively inferior.

Bazaraa and Kirca [2] have implemented a complex scheme involving incomplete tree search and exchange routines to find suboptimal solutions for the FLP. The tree search is modified from the conventional branch and bound procedure as follows:

- Exchange routines are applied to the LAP solutions obtained in the process of computing the Gilmore-Lawler bounds. This is done at all branches or only selected branches depending on the level of the tree. The exchange routine evaluates 2-way (pairwise) exchanges and 2 x 2-way exchanges (simultaneous exchange of locations of two pairs of facilities), iterating between the two until no further improvement is possible.

- A "selective location" rule permits assignment of high-interaction facilities only to central locations and low-interaction facilities only to "off-median" locations.

- "Group assignment" of objects attempts to place sets of facilities with high pairwise interactions close to each other.

Table 2: Results for the Bazaraa-Kirca and Burkard-Bonniger algorithms

PROBLEM	Bazaraa-Kirca[a]		Burkard-Bonniger[b]	
NVR20	1285	2:36	1287	0:50
NVR30	3064	5:20	3072	6:29
STEIN RD	4800	7:46	4822	14:37
STEIN SED	7926	8:27	7987	14:37

[a]Fortran programs on a CDC Cyber 76M

[b]Fortran IV programs on a CDC Cyber 70 model 74-28/CDC 6400

- Artificial upper bounds, depending on the level of the tree, are used to speed up the fathoming process.
- Mirror images, which exist especially in rectangular grid layouts, are removed. (This will be explained in detail in the discussion of the heuristic proposed in this paper.)

This inexact search process is used iteratively by alternating between two branching rules to reduce the dependence on initial partial assignments selected by just one branching strategy. The results obtained by Bazaraa and Kirca are presented in Table 2.

The last heuristic algorithm discussed in this section is due to Burkard and Bonniger [4], who have devised a heuristic which finds cutting planes directly, without Benders' Decomposition, for the Balas and Mazzola [1] linearization. The authors start with a randomly generated solution which is improved by pairwise and triple exchanges. The cutting plane heuristic is then applied to this solution and the new solution is improved by pairwise exchange. The application of one cut and pairwise exchange constitute one iteration of the main algorithm. The method goes through 3n such iterations, after which a new random solution is generated and the process is restarted. The number of restarts used is 10 for $n \leq 20$ and 15 for $n > 20$, and the best result found is used as the final solution. Results from the Burkard and Bonniger paper are presented in Table 2. The solution found in any one run of this algorithm is dependent on the random initial solutions generated during this particular run and hence the results presented for this algorithm are the average results over 10 tests for each individual problem.

Algorithms for solving the facilities layout problem can be placed roughly along some sort of continuum with naive/simplistic heuristics at one end providing only "poor to fair" solutions, to exact methods at the other end providing optimal solutions but requiring prohibitive amounts of computation time. In between, one finds heuristic methods which are more sophisticated than the naive ones and provide better solutions but at some additional cost in computational requirements. Starting from the low-cost, fair/poor solution end and moving to the optimal solution end, one would first find the Liggett algorithm, followed further along by the Burkard-Bonniger and Bazaraa-Kirca algorithms. The algorithm presented in this chapter falls into the same approximate category as the last two, and results will be compared accordingly.

3 A New Hybrid Algorithm

The algorithm proposed in this chapter consists of three parts corresponding to the three points discussed below. In the first part, several partial assignments are generated for use as starting points for a constructive heuristic. In the second part, these starting points are used to construct complete assignments, and the final part attempts to improve the constructed solutions by the application of exchange routines.

3.1 A Constructive Heuristic

The Gilmore-Lawler algorithm provides a way for choosing the next assignment through an extension of the computations required for calculating the lower bound for any partial assignment. It is not necessary to explicitly evaluate additional assignments individually, thus reducing the computational burden.

To calculate a lower bound for a given partial assignment, proceed as follows. Suppose $\mathcal{M}$ is the set of facilities already assigned, and $\mathcal{S}$ is the set of locations that have already been assigned facilities. We begin by evaluating the assignment of an unassigned facility i to a free location j. The incremental cost due to this assignment is:

$$L_{ij} = a_{ij} + \sum_{p \in \mathcal{M}} \left\{ f_{ip} d_{j\rho(p)} + f_{pi} d_{\rho(p)j} \right\} + \sum_{p \notin \mathcal{M}} f_{ip} d_{j\rho(p)} \tag{2}$$

The first two terms in this expression are known exactly and we are required to find

$$\lambda_{ij} = Min \sum_{p \notin \mathcal{M}} f_{ip} d_{j\rho(p)} \tag{3}$$

Define two vectors: F^i is the i^{th} row of $\mathcal{F}$ minus the i^{th} element minus the elements $p \in \mathcal{M}$; D^j is the j^{th} row of $\mathcal{D}$ minus the j^{th} element minus the elements $q \in \mathcal{S}$. Let v_{ij} be the minimum dot (inner) product of F^i and D^j. This minimum is obtained by matching the m^{th} largest element of one vector with the m^{th} smallest element of the other vector. Then

$$\lambda_{ij} = v_{ij} + f_{ii}d_{jj} \tag{4}$$

This value of λ_{ij} is substituted in 2 to obtain L_{ij}. The process is repeated for all pairs (i,j) such that $i \notin \mathcal{M}$ and $j \notin \mathcal{S}$. The optimal solution to the linear assignment problem defined by the matrix of these values of L_{ij} can then be obtained

$$z^* = Min \sum_{i,j} L_{ij}x_{ij} \tag{5}$$

A lower bound on all solutions contained on branches emanating from this node is

$$LB = \sum_{i \in \mathcal{M}} a_{i\rho(i)} + \sum_{i,j \in \mathcal{M}} f_{ij}d_{\rho(i)\rho(j)} + z^* \tag{6}$$

The Assignment Selection Rule

The solution to the LAP defined by equation 5 provides the basis for choosing the next assignment in the constructive heuristic used. The dual variables from the optimal solution can be used to reduce the matrix, i.e. replace each L_{ij} by $L_{ij} - u_i - v_j$. For every $\{i, \rho(i)\}$ element, which will now be zero, find the next smallest element in that row and column, and take their sum. This amount is the *regret* or minimum *additional* cost if assignment $\{i, \rho(i)\}$ is *not* made. Also, this regret plus LB gives us the *alternate cost* of this assignment. Choose the assignment which has the maximum regret, or, equivalently, the maximum alternate cost as the next one to be fixed. Now calculate the lower bound for the augmented assignment and repeat until all but two facilities have been assigned. At this stage, only two completions are possible. Utilizing the known cost component of the lower bound, their values can be calculated easily. The better value and its corresponding assignment are saved. We also refer to this process of starting from a partial assignment and constructing a complete solution as sending a "probe" from the partial assignment; the cost of the complete solution is called the value of the probe.

The probe is modified for problems with dummy facilities (e.g. the Steinberg problems) in the following way. Suppose that there are m dummy facilities. The locations are ranked in increasing order of total distance to and from other locations, and the dummy facilities are assigned to the last m locations in the order. These assignments are made in addition to the ones in the partial assignment used as a starting point for the probe, before beginning the constructive heuristic. The reasoning behind this is that dummy facilities have (by definition) no interaction with any other facility and should be assigned to corner or distant locations.

One run of a constructive heuristic, starting from a single partial solution, is not likely to lead to a good solution. The literature on constructive techniques for the facilities layout problem and experimentation with the heuristic described in this section shows that this is certainly true for the FLP. To obtain better solutions with a constructive procedure, two different strategies can be implemented. The solution provided by the constructive heuristic can be improved by exchange routines, or multiple runs of the constructive procedure can be made. The proposed algorithm uses both the above strategies.

Multiple runs of the constructive heuristic can be made to lead to different solutions in two distinct ways. The first option is to use different starting points. The alternative is to evaluate the possible assignments at the next stage and then choose between them probabilistically, perhaps after weighting the assignments. This causes the heuristic to follow different paths through the decision tree for the problem, culminating in different solutions. This latter method was not as successful as the first in constructing good solutions and the starting points option has been retained in the final design of the algorithm presented in this paper. In the next section, we describe how several starting points can be generated, and in the subsequent section discuss the exchange routines applied to the solutions constructed.

3.2 Starting Points for the Constructive Heuristic

A simple way to get different partial assignments is to choose one facility and assign it to the different locations. For instance, facilities could be ranked in decreasing order of the total flow through them and the first facility could be assigned to all the locations to obtain n different starting points. (We could arrive at n^2 partial assignments by considering the pairing of all facilities with all locations.) It is obvious that some of these assignments would be poor choices since facilities with high interactions (large flows and/or many flows to other facilities) should be located centrally and, conversely, facilities with low interaction levels should be placed in more remote locations. The computational requirements can be reduced, without an unreasonable risk of missing a good solution, by considering only a subset of such pairs. Taking this line of reasoning one step further, more partial assignments could be obtained by using starting points containing two, three or even more assignments. For a problem with n facilities, and m individual assignments in each starting point, there are $\binom{n}{m}$ sets of locations that could be chosen. Each one of these sets could be combined with the m fixed facilities in $m!$ ways. Thus the number of possible partial assignments grows as $m! \times \binom{n}{m}$.

A breadth-first tree search can be suitably modified to serve the purpose of generating several starting points, each one of which contains a relatively good partial assignment. The total number of such partial assignments can be controlled by specifying the lowest (largest index) level of the tree and the rate at which the number of nodes saved grows from one level to the next. It is convenient and computationally less burdensome to fix a facility at each level and evaluate only the assignment of this facility to all free locations. In this algorithm, facilities are ranked in decreasing order of total flow, and the i^{th} facility in the order is fixed at level i of the search tree. This approach has the intuitive appeal of assigning "critical" facilities first, and then fitting less important facilities around them. Let us consider the first level of such a tree. There is only a limited number of central locations to which the first facility (in order) can be reasonably assigned, and the other nodes can be discarded. At level two, under each of the nodes saved at level one, again there is a limited number of free locations which are suitable for the assignment of the second facility.

Following this logic, two versions of restricted breadth-first search trees were developed. Version 1 develops all the nodes at level one and saves the k_1 best. Then all nodes are evaluated under the saved ones and the $k_1 \times k_2$ best are saved. This is repeated until the limited tree has been developed to the desired level, say m; the number of nodes saved at any level i is k_i times the number at the previous level. Version 2 differs from the first in one respect. After saving the k_1 best nodes at level 1, the k_2 best under *each* are saved at level 2. Similarly at any lower level i, the assignments saved are the k_i best under *each* node at the previous level. The total number of partial assignments generated by either version is the same ($k_1 \times k_2 \times \ldots \times k_m$) but version 2 tends to spread out the tree and include nodes that are not so promising in terms of the expected values of their completions, but contain more diverse partial assignments. Such a "fan-shaped" tree provides superior performance within the overall algorithm. The reasons for this will be discussed in detail in the conclusion to this paper.

The probes made from nodes at level m can be augmented by probes that use the partial assignments at level $m-1$. Given that the constructive heuristic and the tree search method use a different criterion for choosing the next assignment, this can lead to different and sometimes better complete solutions. The use of starting points at even higher levels was not found to provide improved solutions.

It was also found that using different values of k_i was an effective strategy, permitting more nodes to be saved at higher levels, while reducing this number at lower levels to save on computation time without any deterioration in solution quality. The number of nodes saved at level one is fixed at four; the number saved at each of the next three levels (two through four) is three; and at levels below that two nodes are saved under each at the higher level. The lowest level of the tree developed, and hence the total number of probes made, is a function of the size of the problem. Information on the choice of these parameters is

Table 3: Number of probes as a function of problem size

Problem size	m	Probes
12 - 20	3	48
21 - 30	4	144
31 - 40	5	324

given in Table 3.

The Evaluation Function

There are two well known evaluation functions for the facilities layout or quadratic assignment problem. The Gilmore-Lawler lower bound is the first and has been explained earlier. Another approach to evaluating a partial assignment was presented by Graves and Whinston [11]. The Graves-Whinston algorithm provides a means of computing an expected value for the completion of any partial assignment, using statistical properties. These expected values can be used in place of lower bounds such as those computed by the Gilmore-Lawler algorithm; the computation time required by the Graves-Whinston method is, however, considerably smaller. Experimental results with the use of these evaluation functions showed that both methods provided equally good solutions, when used as a part of the overall algorithm, but the Graves-Whinston method consumed much less time. Complete runs of the algorithm using the two evaluation functions showed that the time required was 25 to 30% less for the version incorporating the Graves-Whinston method; since the time consumed by the probes and exchange routines is not affected, the advantage over the Gilmore-Lawler method is even greater. Based on these results, the Graves-Whinston method has been retained as the technique used to generate starting points for the probes.

Removing Mirror Images

Consider two complete and distinct assignments (ρ_1 and ρ_2) with the following property:

$$d_{\rho_1(i)\rho_1(j)} = d_{\rho_2(i)\rho_2(j)} \qquad \forall i, j$$

These two assignments have identical objective function values, and are called mirror images [2]. Rectangular grid layouts have several such mirror images. Further, it is possible to identify assignments that, when added to a given partial assignment, would result in assignments that are mirror images. Probes from mirror images result in complete assignments that are themselves mirror

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Figure 1: Mirror images in rectangular grid layouts

images with the same objective function value. Removal of mirror image starting points can be used to reduce the computation time or to increase the number of probes that can be made in the same time. For *rectangular grid layouts*, the assignment of an unassigned facility i to two free locations (p and q) can form mirror image assignments with respect to any given partial assignment (such as that represented by a node of the search tree) if and only if the following two conditions are satisfied:

1. The total distance from p to all other locations is equal to the total distance from q to all other locations.
2. The respective distances from p and q to the *locations occupied* by facilities already assigned are equal.

Consider the 20-facility layout problem with a rectangular grid layout as shown in Figure 1. At level 1 of the tree, the assignment of a given facility to location 8 or 13 would form mirror image partial assignments, as would its assignment to locations 7, 9, 12 or 14. A third group of mirror image assignments involves locations 3 and 18; the fourth group would include locations 2, 4, 17 and 19; a fifth group is formed by locations 6, 10, 11 and 15; and a final group contains locations 1, 5, 16 and 20. Thus we need consider only 6 distinct assignments, of which the last few are not likely to be part of any good solution.

Moving to level 2 of the tree, consider the possible assignments under the node where the first facility has been assigned to location 8 (see Figure 1). Assignment of the second facility to location 7 or 9 would form mirror image partial assignments; locations 12 and 14 would now be a separate group leading to mirror images. Other examples are locations 2 and 4 as one group and locations 17 and 19 as yet another.

The total distance from each facility to all others is computed just once at the beginning of the algorithm, and provides the first quick check for mirror images. If the total distances are equal, the second condition can be checked. Application of conditions (1) and (2) at each level under each node is sufficient to ensure removal of mirror image partial assignments from the set of starting points developed.

3.3 Exchange Improvement

The proposed algorithm incorporates pairwise and triple exchange routines. The complete solutions found by the constructive heuristic are subjected to pairwise exchange. Experimental experience showed that the very high computational burden involved in application of a triple exchange procedure to *all* the improved solutions thus obtained is not justified. This result is corraborated by the experience of other researchers [6,18]. Limited use of a triple exchange routine can, however, be fruitful. In this algorithm, triple exchange is applied to the best solutions available (one for problems with less than 20 facilities; two for problems with 20 to 29 facilities; three for larger problems) after pairwise exchanges have been attempted on all probe solutions at any level. If triple exchange does succeed in improving a solution, pairwise exchange is again attempted on the new solution. Further improvement means that the latest solution can be recycled through alternate application of triple and pairwise exchange, until one of the procedures is unable to find a better solution.

The order used in evaluating exchanges in both routines is facilities ranked in decreasing order of total flows, and the first exchange providing an improvement is carried out. All pairs/triples are evaluated, and if an improvement is made, another iteration is done. Details on the effectiveness of these exchange procedures for the sample problems are provided in the section on computational experience.

3.4 The Algorithm

The program written to execute the algorithm provides the following additional information beside the best solution found and its cost:

- The value of the best assignment found by a probe (before exchange improvement) is stored and reported. This provides a basis for evaluating the effectiveness of the constructive heuristic as a stand-alone solution method.

- The best solution found through the first application of the pair exchange routine to the individual probe solutions is also provided. Comparison of this value with the best solution found permits evaluation of the improvement effected by the triple exchange routine.

Table 4: Computational results for the new heuristic

PROBLEM	1[a]	2[b]	3[c]
NVR12	289	289	0:03.12
NVR15	575	575	0:06.95
ELSHAFEI	8606274	8606274	0:17.84
NVR20	1285	1285	0:22.85
NVR30	3062	3074	5:00.51
STEIN RD	4768[d]	4777	18:39.84
STEIN SED	7926	7926	21:08.54

[a]Best known solution

[b]Solution found by new heuristic

[c]Computation time (mins:secs) on a VAX 8600

[d]New best known solution

The procedure starts by developing the restricted search tree, saving the appropriate number of nodes under each node at the previous level, until level $m-1$ has been reached (see Table 3 for the choice of m). The partial assignments represented by the nodes at this level are used as starting points for the constructive heuristic. The best probe value is updated each time a probe finds a better solution. Each probe solution is subjected to pair exchange improvement and the best solutions found after pair exchange at level $m-1$ are stored. These solutions are then subjected to triple exchange. If an improvement results, pair exchange is again attempted and the procedure goes through another cycle of exchange improvement. The final solution is stored as the best solution found *at this level.* The process is repeated at the lowest level; the final solution *at level* m is compared to the best solution from level $m-1$, and the better of the two gives the best solution found by the hybrid heuristic.

4 Computational Experience

The heuristic of section 3 was used to solve the sample problems mentioned in section 2 and results are presented in Table 4. In column 1 are the best known solutions from the literature; column 2 gives the solution found by the new heuristic; column 3 gives the CPU time required on a VAX 8600 using Fortran programs[1]. The best known solutions for the NVR12, NVR15 and ELSHAFEI problems have been shown to be optimal.

The best known solution reported for the STEIN RD problem was found in

[1]All linear assignment problems were solved using the LAP code of Derigs [5]

17	9	5	6	23	27	22	26	35
2	4	13	12	11	14	21	25	24
18	8	10	1	20	19	32	34	33
16	3	7	15	28	29	30	31	36

Figure 2: New best known solution for the STEIN RD problem (cost = 4768).

Table 5: Comparison of results for chosen algorithms

PROBLEM	Percentage deviation from best known solution		
	Burkard-Bonniger	Bazaraa-Kirca	New Heuristic
NVR20	0.16	0.00	0.00
NVR30	0.33	0.07	0.39
STEIN RD	1.13	0.67	0.19
STEIN SED	0.77	0.00	0.00

the process of experimentation. As mentioned earlier, when there are dummy facilities, they are pre-positioned in corner locations. For the Steinberg problem this means that the two dummy components can be placed in two of locations 1, 9, 28, and 36. There are three ways to do this—use corners on the diagonals, use corners along lengths, or use corners along widths. Each way offers two possibilities that are mirror images of each other. In effect, there are only three choices and we solved the problem for each of the three options. The new best known solution was found by the heuristic when the dummy facilities were placed in corners along a width, and is presented in Figure 2.

Table 5 presents a comparison of the results for the heuristic of section 3 with the results obtained by Burkard-Bonniger and Bazaraa-Kirca. The respective columns give the deviation as a percentage from the best known solution for each problem.

Finally, details on the performance of the algorithm of section 2 are presented in Table 6. Column 1 gives the best probe value before exchange improvement; column 2 gives the best solution found by applying the pair exchange routine to probe solutions; and column 3 gives the best overall solution found by the heuristic. It should be pointed out that the best solution found by applying pair exchange to probe solutions is almost never found from the best probe.

Table 6: Performance details for the new heuristic

PROBLEM	1[a]	2[b]	3[c]
NVR12	291	289	289
NVR15	584	575	575
ELSHAFEI	8728824	8606274	8606274
NVR20	1310	1285	1285
NVR30	3163	3074	3074
STEIN RD	5011	4814	4777
STEIN SED	8417	8117	7926

[a]Best probe value

[b]Best solution found by applying pair exchange

[c]Final solution

5 Analysis of Results and Conclusion

The heuristic proposed in this chapter finds solutions within 0.39% of the best known solutions for all the commonly cited problems in the literature. However, it is instructive to examine its performance in detail to analyse the effectiveness of the component procedures. Toward this end, Table 7 reproduces the information in Table 6 in a different format. The best probe value (column 1), the best solution found by pair exchange (column 2), and the best overall solution (column 3) are given in terms of their percentage deviation from the best known solution.

Column 1 of Table 7 clearly shows that the performance of the constructive heuristic as a stand-alone solution method deteriorates as the problems get larger. Starting with a deviation from the best known solution of less than 1.00% for the 12 facility problem, the deviation steadily increases until it is more than 6.00% for the 36 facility problem. One reason for this is the fact that the Gilmore-Lawler (or Graves-Whinston) evaluation function does not provide an accurate idea about the relative quality of partial assignments. For example, the Gilmore-Lawler bounds at level 1 of a search tree are only about 60% of the best known solution for the NVR30 problem. A similar observation can be made concerning the pair exchange routine (column 2). Pairwise exchange was sufficient to find the optimal/best known solution for the sample problems with 20 or less facilities. For the 30 facility problem, pair exchange was able to find a solution within 0.40% of the best known but for the 36 facility problem using squared euclidean distances, this figure rose to 2.41%. Triple exchange became effective only for the STEIN problems (column 3). With rectilinear distances, triple exchange improved the solution from 4814 to 4777; the subsequent pair exchange did not find a better solution. With squared euclidean distances, the

Table 7: Analysis of performance for the new heuristic

PROBLEM	% deviation from best known solution		
	1[a]	2[b]	3[c]
NVR12	0.69	0.00	0.00
NVR15	1.57	0.00	0.00
ELSHAFEI	1.42	0.00	0.00
NVR20	1.95	0.00	0.00
NVR30	3.30	0.39	0.39
STEIN RD	4.96	0.84	0.19
STEIN SED	6.19	2.41	0.00

[a]Best probe value

[b]Best solution after pair exchange

[c]Final solution

best value after pair exchange was 8117 and triple exchange reduced this to 8029; the final improvement to 7926 was effected by the second pair exchange. In none of the cases was a second run of the triple exchange routine beneficial.

As has already been pointed out, the best solution found by probes generally never gives the best solution after pair exchange. This was true for every one of the sample problems. It was found that pair exchange was most effective on solutions that were not among the best constructed by probes. This fact also explains the superior results obtained by using a tree that was relatively spread out, providing a greater variety of partial (and complete) solutions, with a relatively larger spread of solution quality.

This result has interesting implications for the design of combination heuristics. Most efforts in this area have placed the emphasis on designing better constructive heuristics, with exchange-improvement procedures relegated to the secondary role of achieving small gains, if any, through local optimization. Our results indicate that improvement methods play a pivotal role in finding good solutions, and the more promising approach may be to use simpler and faster methods to construct a larger number of distinct solutions *for the explicit purpose* of applying exchange- improvement techniques. A close look at some competitive methods published in the literature suggests that these algorithms too might be utiiizing the power of improvement methods, but indirectly and without explicit recognition of the fact. The Burkard-Bonniger and Bazaraa-Kirca algorithms also make extensive use of exchange procedures, and these are not necessarily applied only to good solutions. Detailed results are not available for their methods to indicate at what stage and how their best solutions are found, but it is possible that a study would lead to similar conclusions. To sum up, the hybrid algorithm of this chapter presents a relatively easy way to construct fairly good diverse solutions that can be used to exploit the performance of exchange improvement procedures.

References

[1] E. Balas and J.B. Mazzola. *Linearizing Nonlinear 0–1 Programs.* Technical Report Management Sciences Research Report No. 467, Carnegie-Mellon University, Pittsburgh, 1980.

[2] M.S. Bazaraa and O. Kirca. A branch-and-bound-based heuristic for solving the quadratic assignment problem. *Naval Research Logistics Quarterly*, 30, 1983.

[3] M.S. Bazaraa and H.D. Sherali. Benders' partitioning scheme applied to a new formulation of the quadratic assignment problem. *Naval Research Logistics Quarterly*, 27, 1980.

[4] R.E. Burkard and T. Bonniger. *A Heuristic for Quadratic Boolean Programs with Applications to Quadratic Assignment Problems.* Technical Report 81–14, Mathematisches Institut, Universitat zu Koln, 1981.

[5] R.E. Burkard and U. Derigs. *Assignment and Matching Problems: Solution Methods with Fortran Programs.* Volume 184 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin, 1980.

[6] R.E. Burkard and K.H. Stratmann. Numerical investigations on quadratic assignment problems. *Naval Research Logistics Quarterly*, 25, 1978.

[7] A.N. Elshafei. Hospital layout as a quadratic assignment problem. *Operational Research Quarterly*, 28, 1977.

[8] G.K. Gaschutz and J.H. Ahrens. Suboptimal algorithms for the quadratic assignment problem. *Naval Research Logistics Quarterly*, 15, 1968.

[9] J.W. Gavett and N.V. Plyter. The optimal assignment of facilities to locations by branch and bound. *Operations Research*, 14, 1966.

[10] P.C. Gilmore. Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of SIAM*, 10, 1962.

[11] G.W. Graves and A.B. Whinston. An algorithm for the quadratic assignment problem. *Management Science*, 17(7), 1970.

[12] M. Hanan and J.M. Kurtzberg. A review of the placement and quadratic assignment problems. *SIAM Review*, 14(2), 1972.

[13] Bharat K. Kaku and Gerald L. Thompson. Comment on "A perturbation scheme to improve Hillier's solution to the facilities layout problem". *Management Science*, 32(2):251–253, 1986.

[14] Bharat K. Kaku and Gerald L. Thompson. An exact algorithm for the general quadratic assignment problem. *European Journal of Operational Research*, 23(3):382–390, 1986.

[15] L. Kaufman and F. Broeckx. An algorithm for the quadratic assignment problem using Benders' decomposition. *European Journal of Operational Research*, 2, 1978.

[16] A.H. Land. A problem of assignment with inter-related costs. *Operational Research Quarterly*, 14, June 1963.

[17] E.L. Lawler. The quadratic assignment problem. *Management Science*, 9, 1963.

[18] R.S. Liggett. The quadratic assignment problem: an analysis of applications and solution strategies. *Environment and Planning*, 7, 1980.

[19] R.S. Liggett. The quadratic assignment problem: an experimental evaluation of solution strategies. *Management Science*, 27(4), 1981.

[20] C.E. Nugent, T.E. Vollmann, and J. Ruml. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16, Jan–Feb 1968.

[21] C.J. Picone and W.E. Wilhelm. A perturbation scheme to improve Hillier's solution to the facilities layout problem. *Management Science*, 30(10):1238–1249, 1984.

[22] L.P. Ritzman. The efficiency of computer algorithms for plant layout. *Management Science*, 18(5), 1972.

[23] L. Steinberg. The backboard wiring problem: a placement algorithm. *SIAM Review*, 3(1), 1961.